Asking (Question and Answer Platform)

Final Report

Randil Fernando

140161F

**Abstract:**

There are different kind of questions and answer platforms available. University students use internet as the main source of gathering information. Academic related questions occurred to students can categorize according to modules. If there is a question source which is organized according to modules then it will help students to find answers and to exchange their knowledge with fellow students. Asking question and answer platform provide undergraduate students and lecturers a web platform to submit, search, rate questions and answers. Additionally, lecturers can use asking platform to view overall report on questions submitted.

Table of Contents

1. Introduction 4

1.1 Background 4

1.2 Motivation 4

1.3 Importance and Main Purpose 4

1.4 Overview and Summary 5

2. Literature Review 5

2.1 Students 5

2.1.1 Search for questions and answers 5

2.1.2 Submit new question 5

2.1.3 Rate question and answers 5

2.2 Lecturers 5

2.2.1 Update and manage questions and answers submitted 5

2.2.2 Remove unnecessary questions and answers 6

2.2.3 Submit answers 6

2.2.4 View a report of questions submitted 6

2.3 Administrators 6

2.3.1 Update and manage module details 6

2.3.2 Remove unnecessary questions and answers 6

2.3.3 Blacklist unnecessary accounts 6

2.4 System engineer 7

2.4.1 Add new features, repair system break downs and handle errors 7

3. System Models 7

3.1 System Requirement 7

3.2 System Design 9

4. System Implementation 11

4.1 Implementation Procedure 11

4.2 Materials 11

4.3 Main Interfaces 12

5. System Testing and Analysis 14

5.1 Testing Approach 14

5.2 Unit Testing 15

5.3 Aspect Related to Performance, Security and Failures 15

6. Conclusion and Future Work 15

6.1 Conclusion 15

6.2 Future Work 16

7. References 17

# Introduction

## Background

This project is about, providing a question and answer platform for undergraduate students to solve their problems with academic studies and to provide lecturers a way to analyze and review answers submitted by students. As a result of the popularity of the internet most people use it as their main information source and to solve their problems.

There are various type of question and answer platforms available. But there is no proper one which is customized to the usage of undergraduates. Questions occurred to undergraduate students mainly on the modules that they study at university premises. Asking question and answer platform implemented by categorizing questions according to their relevant module. It gives a better user experience to undergraduates. Also, lecturers can create their accounts through the system administrator and it gives them ability to view and analyze questions submitted and to get an overall understanding of students.

Asking platform was built to give undergraduate students a good user experience buy providing them a question and answer platform which was customized according to their needs and to provide lecturers a source which can be used to interact with students.

## Motivation

In this project, I expect to develop a web portal for students where they can submit questions and search answers for problems that occur during their academic studies. Students can search questions and questions are categorized according to module names. This will help a centralized media for university students to exchange their knowledge. System is planned to use crows sourcing as the main information gathering media.

Asking question and answer platform will provide a knowledge base which university students can use with their academic studies and to solve their problems. Also, it enables them a media to exchange their knowledge among their fellow class mates.

## Importance and Main Purpose

Main purpose of the Asking is a Question and Answer Platform which specialize to fulfill the need of platform for undergraduate students submit their questions related to academic studies and exchange their knowledge among class mates. Also, application allow lecturers to submit answers and to get an overall understanding of students by reviewing the questions submitted. The application should be available as a web service and students need to create an account in order to access the service. Lecturer accounts are created and managed through the administration.

Students can submit their questions through the platform by mentioning the module, title and appropriate tags which describe the question. Students can subscribe for modules in order to get notification about newly submitted questions. Lecturer accounts are assigned to modules by the administration. Students and lecturers can submit answers for questions. Rating system is used to prioritize questions and answers.

Further more lecturers can use this system to view a statistical view of questions submitted categorized according to modules and topics. This can use to get an overall understanding about their teaching process and to enhance it

## Overview and Summary

Asking question and answer platform is a web application that service by giving undergraduates a platform to submit, search and to find solutions for their academic related questions and lecturers a platform to solve questions raised by their students and a method to get an overall understanding of students.

# Literature Review

## Students

### Search for questions and answers

Student can search for questions and answers to find answers for their questions. Questions are categorized according to modules. Students can search for questions by providing keywords or they can view and browse questions through modules. By selecting a module end user will get a module details page indicating number of questions submitted. Questions are sorted according to rating. This will prioritize most appropriate answers. While searching answers, answers that was provided by lecturers and answers that was provided by students are displayed in two different areas to give better user experience.

### Submit new question

When a student was unable to find a question, or answer that will solve their problem they can submit new questions providing module name and topic. By indicating module name and topic it will get the attraction of people who have better understanding on those areas. Additionally, while submitting questions question tags can be added to further indicate the related areas.

### Rate question and answers

Students and lecturers can rate questions and answers to change their value. While searching questions and answers most valuable question or answer will be at the top position. This will give a user a proper set of answers. Although students and lecturers are capable of rating questions, lecturer rate is more valuable that the question vote.

## Lecturers

### Update and manage questions and answers submitted

Some questions may be wrong or sometimes there may be a small change to make it a good answer. Lecturer accounts provided an ability to update questions and answers to give students a perfect answer.

### Remove unnecessary questions and answers

Also, sometimes questions may be inappropriate and need to remove otherwise it will decrease the quality of the platform. Only the lecturers provided the ability to remove questions and answer to maintain a proper question database.

### Submit answers

Lecturers can submit answers to the platform. Lecturer answers held major place and it will be at the top. Since answers are categorized into two sections indicating student answers and lecturer answers it is easy for a student to search a proper answer. If all those answers are at a single category then there can be a situation where question with higher rating was not submitted by lectures but there may be lecturer answers. Some times there can be situating where student answers have a better explanation. So, answers are categorized into two categories.

### View a report of questions submitted

Lecturers can view an overall report or a module wise report. A report will contain information about total number of questions, total number of unanswered question and details of questions submitted categorized according to module names. Lecturers can use these reports to get an idea about their modules and to compare them with another module. Also, a module wise report can be generated. Module wise report contain total number of questions submitted, total number of unanswered questions and details of questions submitted categorized according to topics. This will give an idea about students. If there is a topic which has higher number of questions submitted than other areas Then the lecturer can focus more on that area at the next lecturing session.

## Administrators

### Update and manage module details

Asking question and answer platform use module syllabuses to add or update module data into the system. Since those module details not available online this can not automate. Also, when there is a new module added or module changed module details need to update. Them only students can submit questions mentioning module name and topic. Updating topics of the module change how users react to the submitted questions by attracting more users who has a good knowledge on those areas.

### Remove unnecessary questions and answers

Administrators are provided with the functionality to remove unnecessary questions and answers to provide a clear question base and to maintain the quality of the software.

### Blacklist unnecessary accounts

Administrators are provided with functionality to blacklist unnecessary use accounts to maintain the quality and prevent other from corrupting the system.

## System engineer

### Add new features, repair system break downs and handle errors

System engineer will periodically review user feedbacks, and considering admins requests he will add possible new features to the system or upgrade the system. He will also repair the system in break downs or when a possible error occurred in the system.

# System Models

## System Requirement

Functional requirement of asking question and answer platform can described under data to be entered into the system, operations performed by each screen, work-flows performed by the system, system reports or other outputs, system meets applicable regulatory requirements. Asking questions and answer platform has three user types.

Students can search for questions and answers to find answers for their problems. Also, is there is no questions related student can submit new question providing title, module, topic, description and question tags. Then the users who subscribed for the module will get notification. Students can submit new answers for questions submitted by fellow students. Also, rating system improve the search result. Student can rate question and answer to increase or decrease its value. Highest value question or result will be at the top of the search result. Lecturers can submit new questions, rate question and answers, update questions and answers, remove questions and answers and view report. Students and lecturers can subscribe for modules to get notifications. Reporting section held a major role in lecturer accounts. Administrators capable of adding lecturer accounts, manage user accounts and manage questions.

Main non-functional requirements of the system are the speed. Search result should generate within 5-seconds time period. Most of the users interact with the system by searching questions and answers. Therefore, speed of the searching query held a major role in user experience. System should be up for 95% of the day and 5% of the day can be use by system administrators to manage and review system functionalities.



Fig. 1. Use case diagram

In Fig. 1 show that all the use cases related with platform. System is divided into four sub systems; Module management, Report and notification, question and answer management, user management. Fig. 1 will display the user types who can interact with each subsystem.

## System Design



Fig. 2. System architecture diagram

Fig. 2 shows the architecture of the system since asking consist of two parts: one web portal and mobile application. The mobile application is for students and it gives functionalities to search questions and to add new questions and to add answers.

Web application is for all the user types who can interact with the system where it gives search questions, add new question, add new answer, add new module, update module, update question, update answer, remove question and answer, blacklist account. There are three type of users involving with the system: student, lecturer and administrator. Asking question and answer platform provide users with all the functionalities related to a question and answer platform and the advantage of using asking question and answer platform is that it is customized according to undergraduates and questions available inside the platform is specialized to the undergraduate students and categorized according to modules and topics which is easier to browse and search.

Server side of the application use RESTful [1] architecture as the main architecture of the system. Client side interact with web server through JSON [2] objects. All the outputs of the server are JSON object type. Since the server is implemented according to the RESTful [1] architecture token based authorization is used and token is saved on the client side. Client interact with web server with http requests and then that request will invoke a method inside relevant controller finally resulting response with JSON [2] type.

Token based authorization generate the token as the result od the login request and then send back to the client side. Client then save it and send it with every other request it sends. Then the server can authorize request by decrypting token and viewing its content.

Since mongodb [3] used as the persistent database. Model classes that was implemented on the server were mongodb persistent classes. And those classes were implemented using the mongoose [4] library which consist of an object mapper and a database driver. Database schema and the server side classes are identical to each other. Since mongodb [3] is an object based database join operations between mongodb [3] collections are minimized. Most of the collections are isolated with other. Modules, questions, Users are isolated collections while question and answer models has a composition relationship.



Fig. 3. Add new question sequence diagram

Asking question and answer platform use client side rendering and used angular.js [5] as the front-end framework. Therefore, server is accessed only if there is a data retrieval is needed. Fig. 3 shows the add user functionality sequence diagram.

When user requested add new question page client side will generate requested page using reusable components. Since the first step of the above use case does not include data access method no request sent to the server. When there is a data retrieve method then a function inside the angular service is executed and a http request send to the server. Then the URL matcher of the server side will match that request with the relevant route and then the relevant method inside controller will be called.

Controller method will invoke a query on model class which is a persistent class and the object relational mapper will query the database and retrieve the result. Then that result will send back to the server and view will update according to the retrieved data.

Since angular.js provide mapping mechanism with html document object model it is not need to directly manipulate html document.

# System Implementation

## Implementation Procedure

Asking question and answer platform is a web portal. Back-end of the application is implemented using node.js [6] and express.js [7] framework. As the persistent layer mongodb [3] was used. Angular.js [5] used to front-end development.

Software engineering process followed Rational Unified Process methodology. Construction phase consist of four iterations and at the end of each iteration testing and bug fixing implemented. Implementation of functional requirements divided within second and third iteration of the construction phase while first iteration consists of architecture implementation and configuration and fourth iteration consist of security implementation and overall bug fixing.

Back-end of the system used node.js to develop system. JavaScript used as the programming language. Express.js [7] which is a minimal framework used to handle routes, middleware. Express.js is a minimal framework and all the route points need to implement manually hence it is more flexible than other frameworks. Asynchronous behavior of JavaScript increased the software performance and architectures relevant to that behavior was used. RESTful [1] architecture used to develop the server hence all the server requests and responses passed through JSON [2] type. Authentication process was implemented using a token base authentication. Token used to authenticate was JSON [2] web token which is called JWT [8]. Token used single time interaction with database and the token is then saved in the client side. When there is a request the token must send with it. No session was maintained on the server. User email confirmation used SMTP protocol to send emails. Nodemailer [9] library used to send emails.

Front-end development used angular.js [5]. Model-view-viewmodel architecture used at the front-end. Asking platform used client side rendering and used reusable components. There are two routers in the application; view router and REST [1] API router. View router is capable if switching views with the user interactions. The application was a single page web application which consist only one page and generating different views by changing components.

Testing process used jasmine [10] framework to implement test cases. Karma [10] test runner used to automate the unit testing process and protractor used automate end to end testing process which involved testing REST [1] API. Selenium IDE [10] used to automate the user interface testing.

## Materials

Syllabuses provided with module used to add and update module information. Module data are stored in the mongodb database according to the schema created for module model.

## Main Interfaces

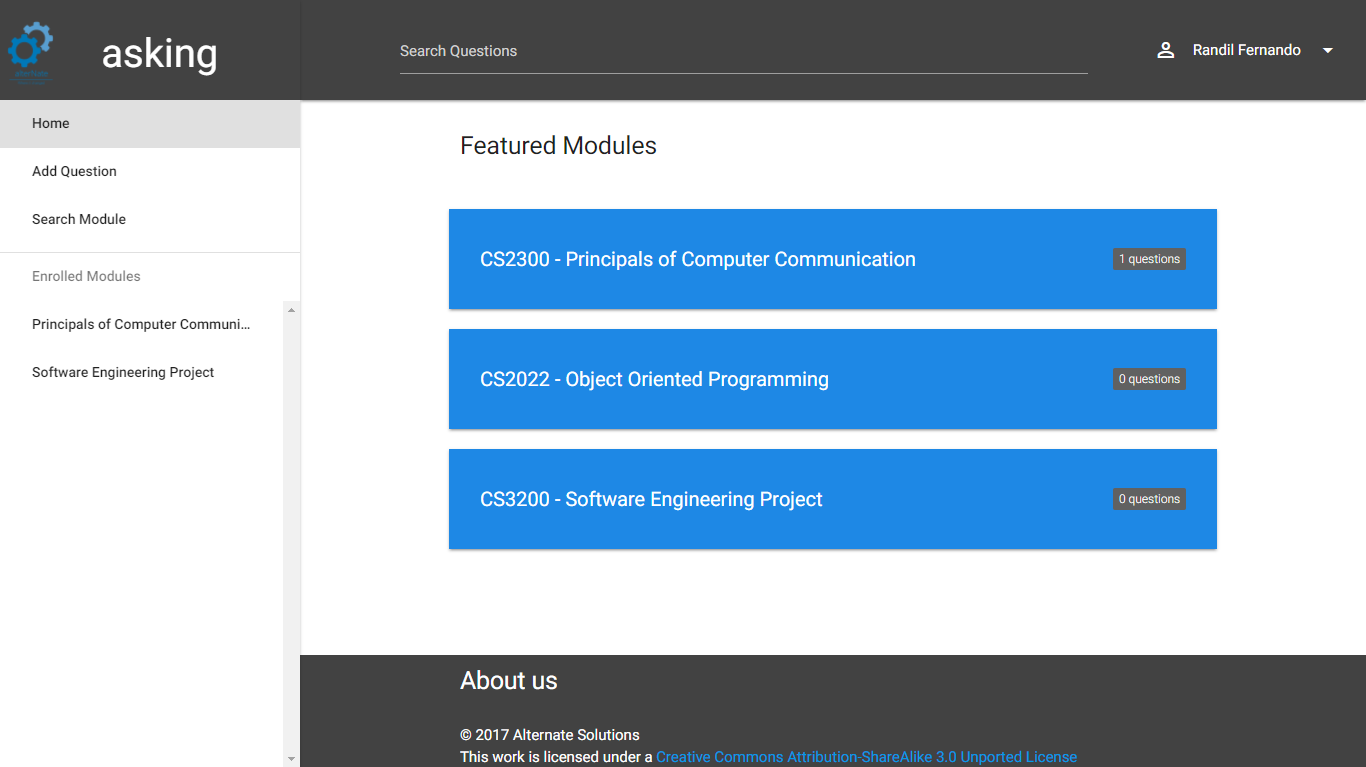


Fig. 4. Home page

As shown in Fig. 4 home page contain sidebar which will give the functionalities available to the logged-on user. Featured modules list contains list of modules which has highest number of questions submitted which is organized according to the descending order. Functionalities available to users will change with the user type hence options available in the sidebar will slightly change with the user. Featured module section is common to all user accounts.

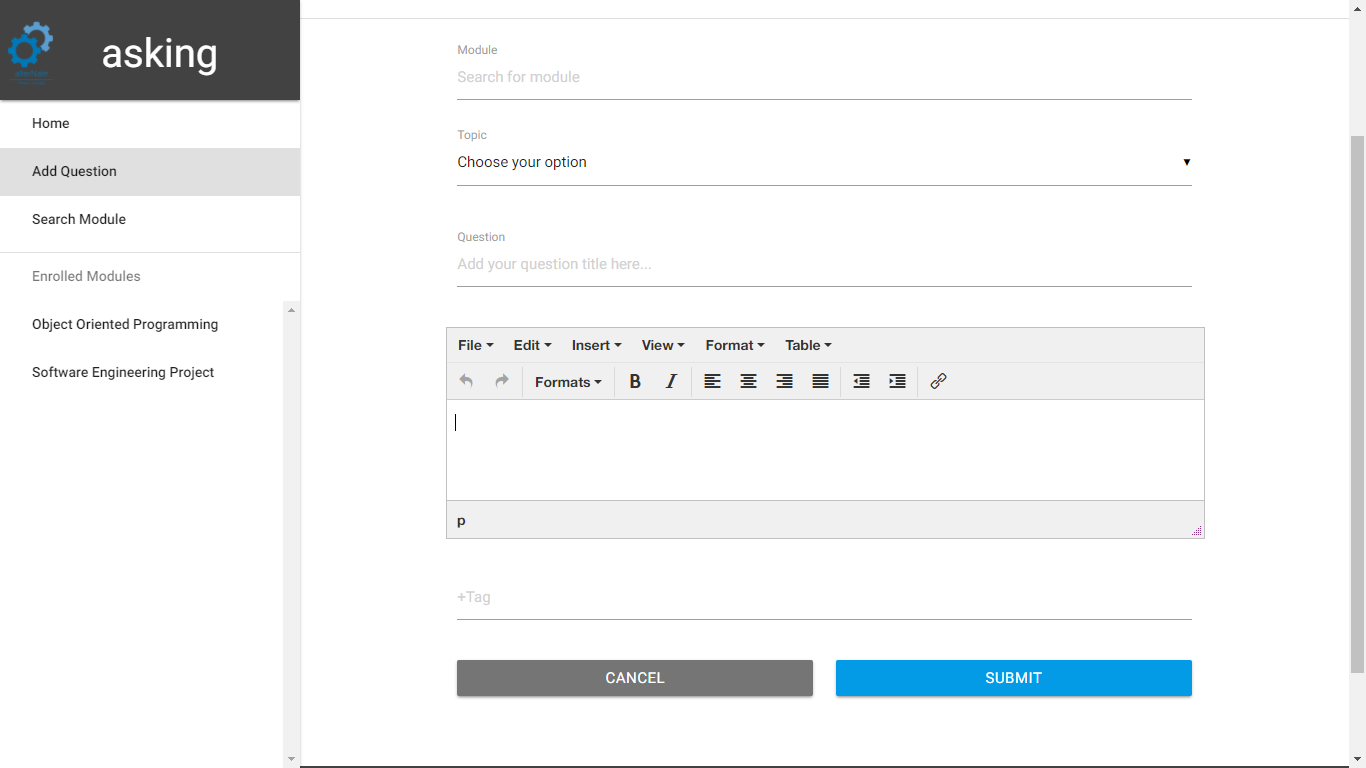


Fig. 5. Add new question page

As show on Fig. 5 students can add new questions to the system. Module can be searched through a search bar and then number of matched module names will be displayed to the user to choose from. After the selection topics will be updated according to the selected module and then the user can select a topic which correctly identify the question. For further identification tags can be added. And then the user can submit new question to the system. Since the reporting section extract data using module code and topic it is essential to add a module and a topic to newly submitting questions. Question title is a must and questions description is an optional attribute.

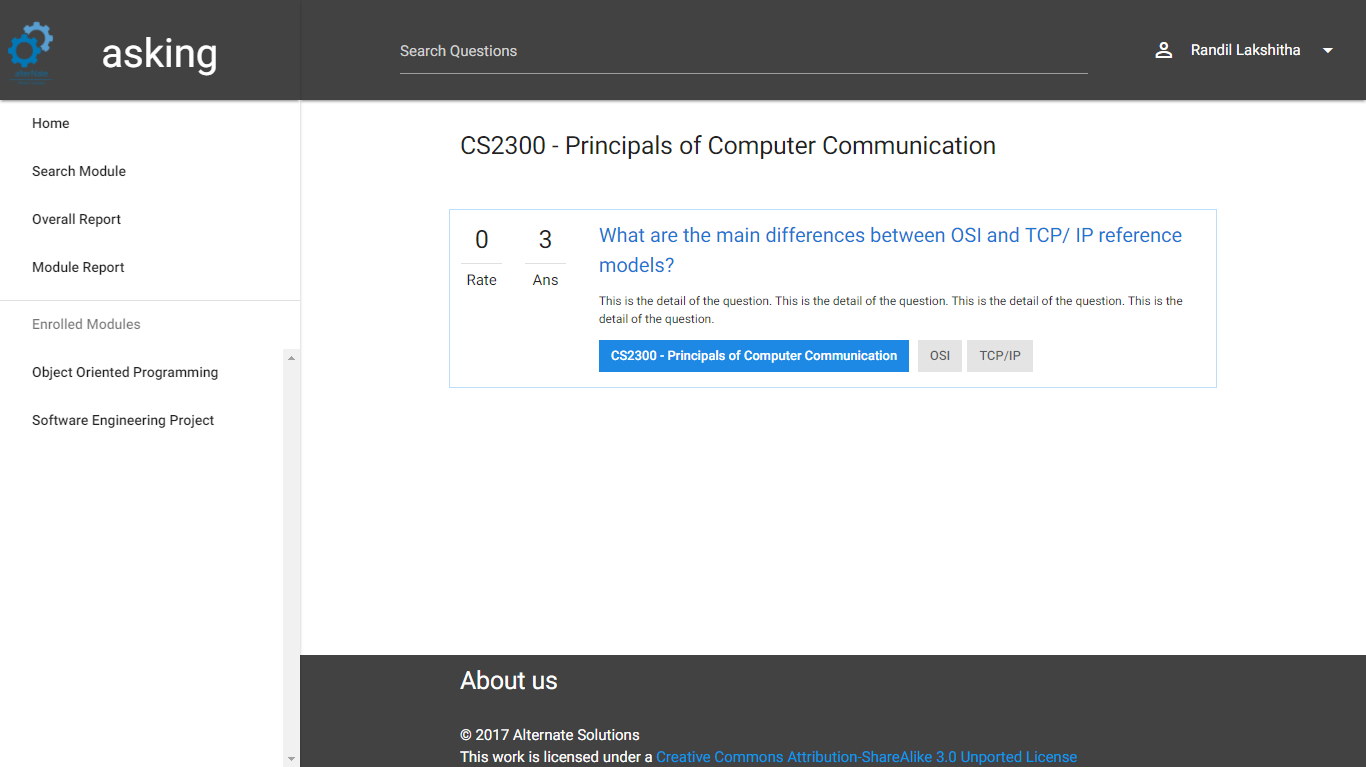


Fig. 6. Search questions page

As shown in the Fig. 6 student, lecturer can search for questions buy providing keywords. Search result will be organized according to the number of words matched with the questions. When user enter a search, prepositions are removed from it and then search process will further. Also, user can browse questions through modules.

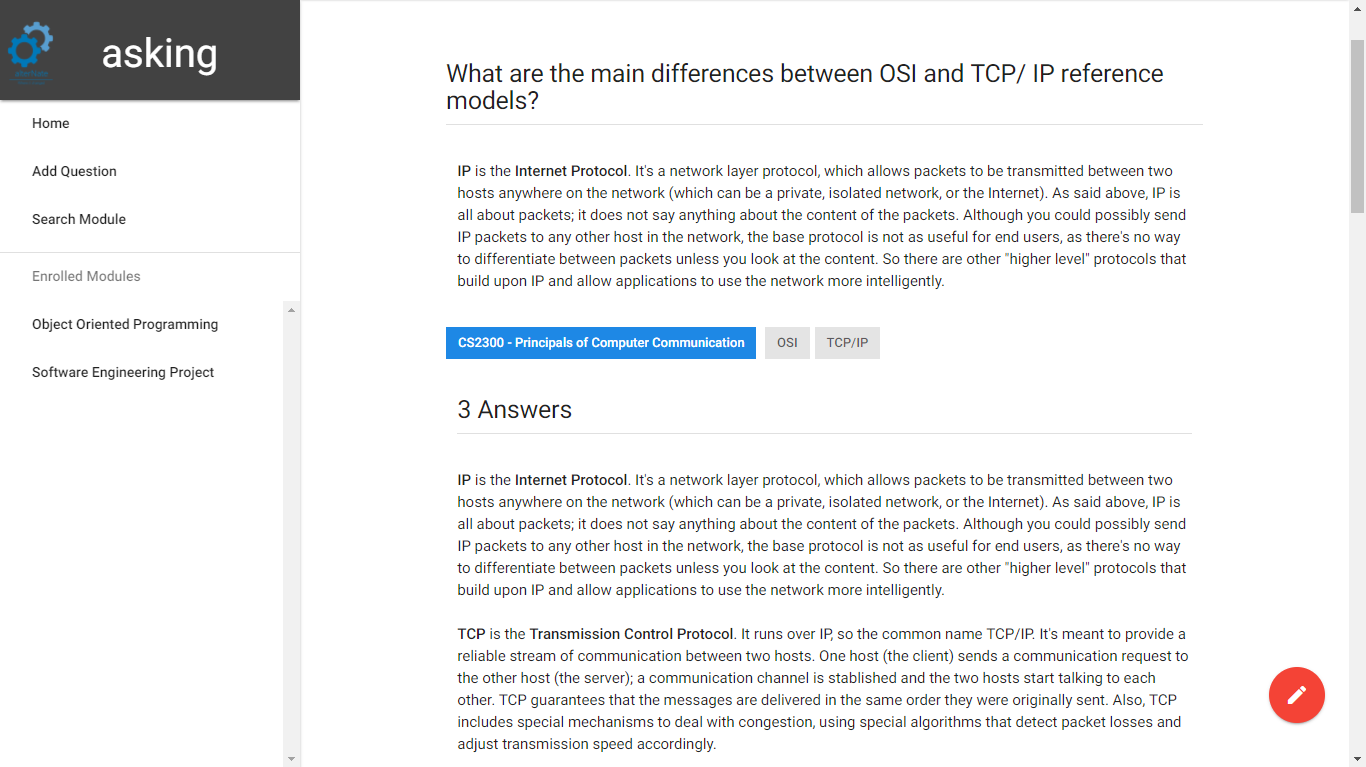


Fig. 7. View question page

As shown in the Fig. 7 when a user selected a question detail page will be displayed with the question description and with the submitted answers. Students and lecturers can add new answers to the answer list. Lecturers can edit question details and also the answers. Students able to update questions and answers which they have been submitted. Lecturers have privilege to edit and remove any question or answer. Administrators has the ability to remove questions or answers.

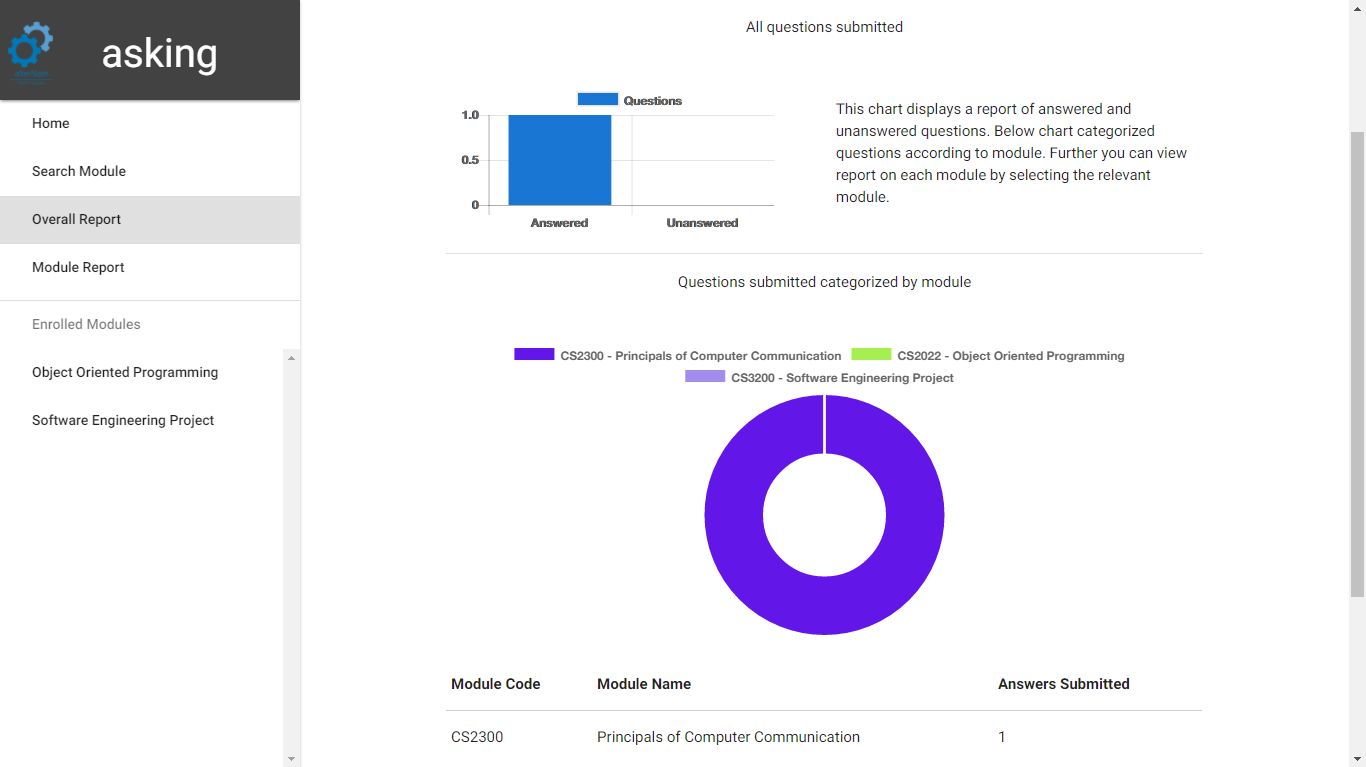


Fig. 8. View report page

As shown in the Fig. 8 lecturers can generate an overall report. Report displays the number of questions submitted to the system with the number of answered questions to get an overall understanding. The bar chart shown above will display a comparison between answered and unanswered questions. Also, the report will display a detail of questions submitted to the system categorized by modules. Donut chart show in the figure #number will display report categorized by module names.

# System Testing and Analysis

## Testing Approach

A test approach is the test strategy implementation of a project, defines how testing would be carried out. Test approach has two techniques

**Proactive**: An approach in which the test design process is initiated as early as possible in order to find and fix the defects before the build is created.

**Reactive**: An approach in which the testing is not started until after design and coding are completed.

Main testing approaches asking question and answer platform use are unit testing, function testing, user interface testing and security and access control testing.

## Unit Testing

A select number of methods will be tested in a couple of classes with black and white box testing to ensure that they function correctly. Asking question and answer platform based on front end angular.js [5] and back end node.js [6] while JavaScript on both front-end and back-end. There are lots of frameworks that provide unit testing functionality to node.js [6] and angular.js [5]. Since both the front-end and back-end use JavaScript same framework can be used to implement test cases.

Jasmine [10] framework used as the unit testing framework it is a behavior driven development style which is focusses of the language used.

Karma [10] which is an automated test runner can be used to automate unit testing of the front-end. Also, protractor [12] which can be used to automate the end to end testing process. On the server side, jasmine [10] framework can be used to automate the testing process.

## Aspect Related to Performance, Security and Failures

Searching and browsing questions is the main function of the system. Therefore, performance of searching and browsing questions will directly affect the user experience. Slow network speeds will decrease the user experience. Since the system is depend on client side rendering initial loading time of the system is compared to application with server side rendering.

Since the application uses JSON web tokens [8] login token always stored at client side. If user has lost the device which has the stored token then the user cannot remove that device through the server. Only way to expire that token is when the time pass or server need to change the secret key which uses to encrypt the token. If the security key is changed then every user who has a token which was encrypted with the previous secret need to log to the system again.

Node.js [6] is running using a single thread if there is a failure happen whole process will be delayed. Nginx [13] can be used to divide load into multiple instances if the server hence a solution to above problem.

# Conclusion and Future Work

## Conclusion

Asking platform is a web application which provide question and answer platform for undergraduate students and lecturers which is designed and customized to suite university academic related questions. Main purpose of the asking platform is to provide a customized platform which categorized questions according to modules and to provide lecturers a source to view report on questions occurred within students and to get an overall understanding.

Asking platform provide students a media to find answers to their academic related problems and to exchange their knowledge with fellow students. Students can search questions, browse questions which is organized according to modules. Lecturers have higher privilege than students hence more functionalities. Lecturer can modify question and answers, view report of question submitted. Students and lecturers can rate questions and answers. Rating will increase the quality of search results.

System administrator is capable of remove questions and answers, create lecturer accounts, manage user accounts. System administrator provided with these functionalities to improve and to maintain the quality of the system. Also, the lecturer is provided with updating question and answers to maintain the quality of answers.

Another important thing is subscription system. Student and lecturers can subscribe for modules to get notifications.

## Future Work

When there is huge number of users accessing the system sometimes there can be situations where more than one user viewing and modifying the same question or adding answer to the same question. Since the system is not syncing sometimes there can conflicts between users. Therefore, real time implementation of the system is needed and socket.io can be used to introduce real time feature to the system.

Some times there may be situations where users can connect as a group and solve a question. Messaging system with group support can increase the usability of the system. Groups can be created automatically using subscription data. And then if there are signed in users inside that group then a messaging session can be started. Since the group made with users who has a good knowledge on that field it will produce good solutions.

# References

|  |  |
| --- | --- |
| [1] | D. K. Barry, "Service Architecture," [Online]. Available: http://www.service-architecture.com/articles/web-services/representational\_state\_transfer\_rest.html. [Accessed 12 May 2017]. |
| [2] | "Introducing JSON," [Online]. Available: http://www.json.org/. [Accessed 12 May 2017]. |
| [3] | MongoDB, Inc, "What is mongoDB," MongoDB, Inc, [Online]. Available: https://www.mongodb.com/what-is-mongodb. [Accessed 5 May 2017]. |
| [4] | LearnBoost, "Mongoose ODM," LearnBoost, [Online]. Available: http://mongoosejs.com/. [Accessed 12 May 2017]. |
| [5] | Google Inc., "Angular," Google Inc., [Online]. Available: https://angular.io/. [Accessed 5 May 2017]. |
| [6] | Node Foundation, "Node.js," Node.js Foundation, [Online]. Available: https://nodejs.org/en/about/. [Accessed 5 May 2017]. |
| [7] | Node.js Foundation, "Express.js," Node.js Foundation, [Online]. Available: https://expressjs.com/. [Accessed 5 May 2017]. |
| [8] | Auth0, "JSON web tokens," Auth0, [Online]. Available: https://jwt.io/. [Accessed 12 May 2017]. |
| [9] | A. Reinman, "Nodemailer," [Online]. Available: https://nodemailer.com/about/. [Accessed 12 May 2017]. |
| [10] | PivotalLabs, "Jasmine Documentation," PivotalLabs, [Online]. Available: https://jasmine.github.io/. [Accessed 12 May 2017]. |
| [11] | Google Inc., "Karma - Spectacular test runner for javascript," Google Inc., [Online]. Available: https://karma-runner.github.io/1.0/index.html. [Accessed 5 May 2017]. |
| [12] | Selenium HQ, "Selenium IDE - Plugins," Selenium HQ, [Online]. Available: http://www.seleniumhq.org/projects/ide/. [Accessed 5 May 2017]. |
| [13] | "Protractor, End to end testing for angular.js," [Online]. Available: http://www.protractortest.org/#/. [Accessed 5 May 2017]. |
| [14] | NGINX Inc., "NGINX | High Performance Load Balancer, Web Server," NGINX Inc., [Online]. Available: https://www.nginx.com/. [Accessed 5 May 2017]. |
| [15] | R. Fernando, "Asking Project Proposal," Alternate Solutions, 2017. |
| [16] | R. Fernando, "Software Design Specification," Alternate Solutions, 2017. |